

LANGKAH-LANGKAH PRAKTIKUM 2

1. Langkah pertama, membuat database, menggunakan database tersebut dan membuat tabel didalamnya.

```
CREATE DATABASE praktikum2;
```

```
USE praktikum2;
```

```
CREATE TABLE Mahasiswa (  
    NIM VARCHAR(10) NOT NULL PRIMARY KEY,  
    nama VARCHAR(50) NOT NULL,  
    jenis_kelamin VARCHAR(20) NOT NULL,  
    tgl_lahir DATE NOT NULL,  
    jalan VARCHAR(100) NOT NULL,  
    kota VARCHAR(50) NOT NULL,  
    kode_pos VARCHAR(10) NOT NULL,  
    no_hp VARCHAR(20) NOT NULL,  
    kode_dosen VARCHAR(10) NOT NULL  
);
```

```
CREATE TABLE Dosen (  
    kode_dosen VARCHAR(10) NOT NULL PRIMARY KEY,  
    nama VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Matakuliah (  
    kode_matakuliah VARCHAR(10) NOT NULL PRIMARY KEY,  
    nama VARCHAR(50) NOT NULL,  
    SKS INT NOT NULL  
);
```

```
CREATE TABLE jadwal_mengajar (  
    kode_matakuliah VARCHAR(10) NOT NULL PRIMARY KEY,  
    kode_dosen VARCHAR(10) NOT NULL PRIMARY KEY,  
    hari VARCHAR(10) NOT NULL,  
    waktu VARCHAR(10) NOT NULL,  
    nama VARCHAR(50) NOT NULL  
);
```

```

kode_dosen VARCHAR(10) NOT NULL,
kode_matakuliah VARCHAR(10) NOT NULL,
hari ENUM('Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu') NOT NULL,
jam TIME NOT NULL,
ruang VARCHAR(20) NOT NULL,
PRIMARY KEY (kode_dosen, kode_matakuliah, hari, jam)
);

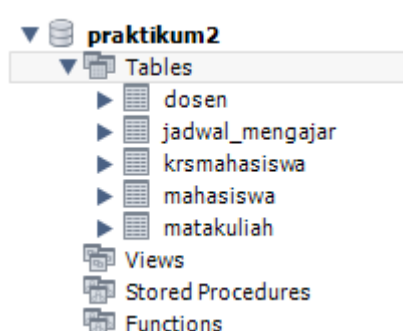
```

```

CREATE TABLE KRSMahasiswa (
    NIM VARCHAR(10) NOT NULL,
    kode_matakuliah VARCHAR(10) NOT NULL,
    kode_dosen VARCHAR(10) NOT NULL,
    semester VARCHAR(10) NOT NULL,
    nilai INT NOT NULL,
    PRIMARY KEY (NIM, kode_matakuliah, kode_dosen)
);

```

OUTPUT:



2. Langkah kedua memasukan semua data kedalam tabel.

```

INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`,
`jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223344', 'Ari Santoso',
'Laki-laki', '1998-10-12', 'Jl. Cempaka', 'Bekasi', '17121', '089911122331', '554321');

```

```

INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`,
`jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223345', 'Ario Talib', 'Laki-
laki', '1999-11-16', 'Jl. Lemah abang', 'Cikarang', '17534', '089911122332', '554321');

```

```

INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`,
`jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223346', 'Dina Marlina',
'Perempuan', '1997-12-01', 'Jl. Juanda', 'Karawang', '41311', '089911122333', '554321');

```

```
INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`, `jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223347', 'Lisa Ayu', 'Perempuan', '1996-01-02', 'Jl. Achmad Yani', 'Bekasi', '17124', '089911122334', '554321');
```

```
INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`, `jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223348', 'Tiara Wahidah', 'Perempuan', '1980-02-05', 'Jl. Raya Ciselang', 'Bekasi', '17142', '089911122335', '554321');
```

```
INSERT INTO `praktikum2`.`mahasiswa` (`NIM`, `nama`, `jenis_kelamin`, `tgl_lahir`, `jalan`, `kota`, `kode_pos`, `no_hp`, `kode_dosen`) VALUES ('11223349', 'Anton Sinaga', 'Laki-laki', '1988-03-10', 'Jl. Ki Balok', 'Cikarang', '17530', '089911122336', '554321');
```

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
...	11223344	Ari Santoso	Laki-laki	1998-10-12	Jl. Cempaka	Bekasi	17121	089911122331	554321
	11223345	Ario Talib	Laki-laki	1999-11-16	Jl. Lemah ...	Cikarang	17534	089911122332	554321
	11223346	Dina Marlina	Perempuan	1997-12-01	Jl. Juanda	Karawang	41311	089911122333	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321
	11223349	Anton Sinaga	Laki-laki	1988-03-10	Jl. Ki Balok	Cikarang	17530	089911122336	554321

3. Langkah ketiga mengubah data lahir mahasiswa Bernama Ari menjadi 1979-08-31

```
UPDATE `praktikum2`.`mahasiswa` SET `tgl_lahir` = '1979-08-31' WHERE (`NIM` = '11223344');
```

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223344	Ari Santoso	Laki-laki	1979-08-31	Jl. Cempaka	Bekasi	17121	089911122331	554321
	11223345	Ario Talib	Laki-laki	1999-11-16	Jl. Lemah ...	Cikarang	17534	089911122332	554321
	11223346	Dina Marlina	Perempuan	1997-12-01	Jl. Juanda	Karawang	41311	089911122333	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321
	11223349	Anton Sinaga	Laki-laki	1988-03-10	Jl. Ki Balok	Cikarang	17530	089911122336	554321

4. Langkah ke-empat Tampilkan satu baris data yang diubah tadi dengan nama Ari saja

```
SELECT * FROM mahasiswa WHERE nama = 'Ari Santoso';
```

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223344	Ari Santoso	Laki-laki	1979-08-31	Jl. Cempaka	Bekasi	17121	089911122331	554321

5. Langkah kelima, menghapus mahasiswa Bernama Dina

DELETE FROM Mahasiswa WHERE NIM = '11223346';

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223344	Ari Santoso	Laki-laki	1979-08-31	Jl. Cempaka	Bekasi	17121	089911122331	554321
	11223345	Ario Talib	Laki-laki	1999-11-16	Jl. Lemah ...	Cikarang	17534	089911122332	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321
	11223349	Anton Sinaga	Laki-laki	1988-03-10	Jl. Ki Balok	Cikarang	17530	089911122336	554321

6. Langkah ke-enam, menampilkan record atau data yang tanggal kelahirannya lebih atau sama dengan 1996-1-2

SELECT * FROM Mahasiswa WHERE tgl_lahir >= '1996-01-02';

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223345	Ario Talib	Laki-laki	1999-11-16	Jl. Lemah ...	Cikarang	17534	089911122332	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321

7. Langkah ke-tujuh menampilkan semua Mahasiswa perempuan yang berasal dari Bekasi.

SELECT * FROM Mahasiswa WHERE jenis_kelamin = 'Perempuan' AND kota = 'Bekasi';

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321

8. Langkah ke-delapan, menampilkan Mahasiswa laki-laki yang berasal dari Bekasi atau mahasiswa perempuan yang berumur lebih dari 22 tahun.

SELECT * FROM Mahasiswa

WHERE (jenis_kelamin = 'Laki-laki' AND kota = 'Bekasi')

OR (jenis_kelamin = 'Perempuan' AND tgl_lahir <= DATE_SUB(CURDATE(), INTERVAL 22 YEAR));

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
▶	11223344	Ari Santoso	Laki-laki	1979-08-31	Jl. Cempaka	Bekasi	17121	089911122331	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321

9. Langkah ke-sembilan, hanya menampilkan nama dan alamat saja dari tabel.

```
SELECT nama, jalan, kota  
FROM Mahasiswa ;
```

OUTPUT:

	nama	jalan	kota
►	Ari Santoso	Jl. Cempaka	Bekasi
	Ario Talib	Jl. Lemah ...	Cikarang
	Lisa Ayu	Jl. Achmad...	Bekasi
	Tiara Wahidah	Jl. Raya Ci...	Bekasi
	Anton Sinaga	Jl. Ki Balok	Cikarang

10. Langkah ke-sepuluh menampilkan data mahasiswa yang berurutan berdasarkan nama

```
SELECT * FROM Mahasiswa ORDER BY nama ASC;
```

OUTPUT:

	NIM	nama	jenis_kelamin	tgl_lahir	jalan	kota	kode_pos	no_hp	kode_dosen
►	11223349	Anton Sinaga	Laki-laki	1988-03-10	Jl. Ki Balok	Cikarang	17530	089911122336	554321
	11223344	Ari Santoso	Laki-laki	1979-08-31	Jl. Cempaka	Bekasi	17121	089911122331	554321
	11223345	Ario Talib	Laki-laki	1999-11-16	Jl. Lemah ...	Cikarang	17534	089911122332	554321
	11223347	Lisa Ayu	Perempuan	1996-01-02	Jl. Achmad...	Bekasi	17124	089911122334	554321
	11223348	Tiara Wahidah	Perempuan	1980-02-05	Jl. Raya Ci...	Bekasi	17142	089911122335	554321

Pertanyaan:

Apa bedanya penggunaan BETWEEN dan penggunaan operator >= dan <=?

Jawab:

BETWEEN, operator >=, dan <= adalah operator perbandingan dalam SQL yang digunakan untuk membandingkan nilai di antara dua nilai atau lebih.

BETWEEN digunakan untuk memeriksa apakah suatu nilai berada di antara dua nilai atau tidak. Misalnya, kita dapat menggunakan BETWEEN untuk mencari semua nilai dalam suatu tabel yang berada di antara dua tanggal tertentu.

Operator >= digunakan untuk memeriksa apakah suatu nilai lebih besar dari atau sama dengan nilai lain. Misalnya, kita dapat menggunakan operator >= untuk mencari semua nilai dalam suatu tabel yang lebih besar dari atau sama dengan suatu nilai tertentu.

Operator <= digunakan untuk memeriksa apakah suatu nilai lebih kecil dari atau sama dengan nilai lain. Misalnya, kita dapat menggunakan operator <= untuk mencari semua nilai dalam suatu tabel yang lebih kecil dari atau sama dengan suatu nilai tertentu

Dalam penggunaannya, perbedaan antara ketiganya adalah bahwa BETWEEN digunakan untuk memeriksa apakah suatu nilai berada di antara dua nilai tertentu, sedangkan operator >= dan <= digunakan untuk memeriksa apakah suatu nilai lebih besar atau lebih kecil dari suatu nilai tertentu.

Kesimpulan:

Berdasarkan pembuatan database yang terdiri dari beberapa tabel yaitu Mahasiswa, Dosen, Matakuliah, Jadwal_Mengajar, dan KRSMahasiswa, dapat disimpulkan bahwa database ini dirancang untuk mengelola informasi mengenai mahasiswa dan mata kuliah yang diambil oleh mahasiswa. Tabel Mahasiswa menyimpan informasi mengenai NIM, nama, jenis kelamin, tanggal lahir, alamat, nomor telepon, dan kode dosen. Tabel Dosen menyimpan informasi mengenai kode dosen dan nama dosen. Tabel Matakuliah menyimpan informasi mengenai kode mata kuliah, nama mata kuliah, dan jumlah SKS. Tabel Jadwal_Mengajar menyimpan informasi mengenai jadwal mengajar dosen pada suatu mata kuliah, termasuk hari, jam, dan ruang. Terakhir, tabel KRSMahasiswa menyimpan informasi mengenai mata kuliah yang diambil oleh mahasiswa beserta dosen pengampu dan nilai.

Dengan adanya database ini, dapat memudahkan pengelolaan data mengenai mahasiswa dan mata kuliah yang diambil oleh mahasiswa. Dosen dapat dengan mudah mengakses jadwal mengajar mereka, sedangkan mahasiswa dapat dengan mudah mengakses informasi mengenai mata kuliah yang diambil dan nilai yang diperoleh. Selain itu, penggunaan database ini dapat meningkatkan efisiensi dan akurasi pengolahan data, menghindari duplikasi data, dan memudahkan pencarian dan pengambilan data yang diperlukan.